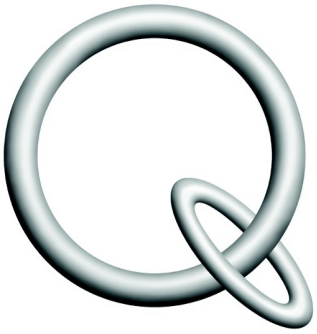
WINCON WEB DAEMON

USER'S MANUAL



QUANSER

INNOVATE. EDUCATE.

How to contact Quanser:

	+1 (905) 940-3575	Telephone
	+1 (905) 940-3576	Facsimile
	80 Esna Park Drive, # 1-3 Markham, ON Canada L3R 2R6	Mail
	http://www.quanser.com	Web
	mailto://info@quanser.com	General information

WinCon Web Daemon User's Manual

MATLAB is a registered trademarks of The MathWorks, Inc.
Other brands and their products are trademarks or registered trademarks of their respective holders and should be noted as such.

© 2005 Quanser Inc.
All rights reserved. This work may not be translated or copied in whole or in part without the written permission of the copyright holder, except under the terms of the associated software license agreement. No part of this manual may be photocopied or reproduced in any form.
The use of general descriptive names, trade names, trademarks, etc. in this publication, even if the former are not especially identified, is not to be taken as a sign that such names as understood by the Trade Marks and Merchandise Marks Act, may accordingly be used freely by anyone.

Printed in Canada.

Table of Contents

Introduction **1**

Overview **2**

Connecting A System 2

NEES Connect Block 5

NEES Signal Block 6

NEES Scope Block 7

Software Prerequisites **8**

Software Requirements Checklist 8

Running the Software **9**

Configuration From the Application 9

Using the Defaults.xml File 12

Opening a Configuration File 14

Saving a Configuration File 14

Creating a New Defaults.xml file 15

List of Figures

Figure 1: Overview of the required software components..... 1
Figure 2: Components required for stream data from a Simulink diagram.....3
Figure 3: Components required to send data to the Simulink diagram.....4
Figure 4: Block parameters for the NEES Connect block.....5
Figure 5: Parameters for the NEES Signal block..... 6
Figure 6: Parameters for the NEES Scope block..... 7
Figure 7: Opening window of the WinCon Web Daemon.....9
Figure 8: The RBNB connection configuration dialog..... 10
Figure 9: WinCon connection configuration dialog.....11
Figure 10: The "Daemon running" message displayed when connected to Data Turbine..... 12

INTRODUCTION

Quanser's WinCon software allows the user the unique ability to design hardware- in-the-loop systems. Integrated with MATLAB's Simulink design environment, a control system may be graphically designed. This graphically designed system may then be run in real-time, on a Windows based machine. This ability to rapidly design and execute code in a development environment makes WinCon extremely useful in the rapid prototype and development fields.

By modifying the control system to include blocks for streaming of data, this data may then be streamed to the Internet. By streaming this data to a centralized location, or repository allows many users to visit this location and retrieve and modify the data.

Creare offers a product called RBNB Data Turbine which is a dynamic data server. The Data Turbine acts like a distributed network file system for data. In order to facilitate the sharing of data and collaboration between labs, the WinCon Web Daemon was developed. This software acts as a bridge between WinCon and the Data Turbine. The controller running in real-time under WinCon's control streams data to the WinCon Web Daemon. The Web Daemon in turn passes the data on to the Data Turbine. Similarly, data posted to the Data Turbine may be retrieved and presented to the running controller as a parameter value. These software components permit the sharing of data and open up the possibility of remotely sharing and collaborating on projects.

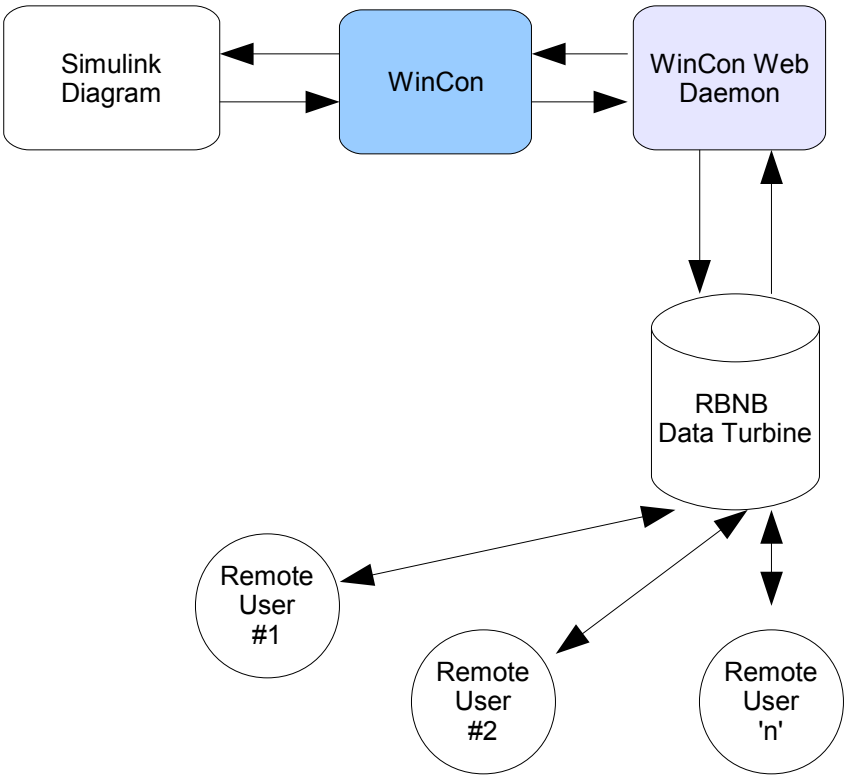


Figure 1: Overview of the required software components.

OVERVIEW

The complete system is composed of many individual components. These components are discussed here to familiarize the user with them. To help demonstrate the concepts involved, an example will be used. The example is a shake table system capable of replaying earthquake data to test structural models.

The basic design process involves the following steps:

1. **Designing the control system**. This will result in a graphically designed controller, capable of controlling the motion of the shake table.
2. **Build the control system**. Building the system is similar to compiling code. In the graphical environment, each of the blocks actually represents instructions or code. Building the control system creates an executable (a type of program) which WinCon can run.
3. **Run the control system**. The system is run as a program under the control of WinCon. WinCon is responsible for making sure the controller runs in real-time.

The process described above is an overly simplified view used solely for illustrative purposes.

Connecting A System

The first step to publishing data to the online repository is to create a Simulink diagram. The Simulink diagram is what controls the system. Once the control system is designed, blocks may be added to stream specific variables to the Internet. In the example used, the control system for the shake table is in a block called **Shake Table Control System.** Within this block are the functions that control the shake table. For our purposes, this is not important to discuss here since we are focusing on connecting a previously built system to the outside world. We wish to display the **Displacement** variable which represents the position of the shake table stage, and we will use a NEES Scope block to do this.

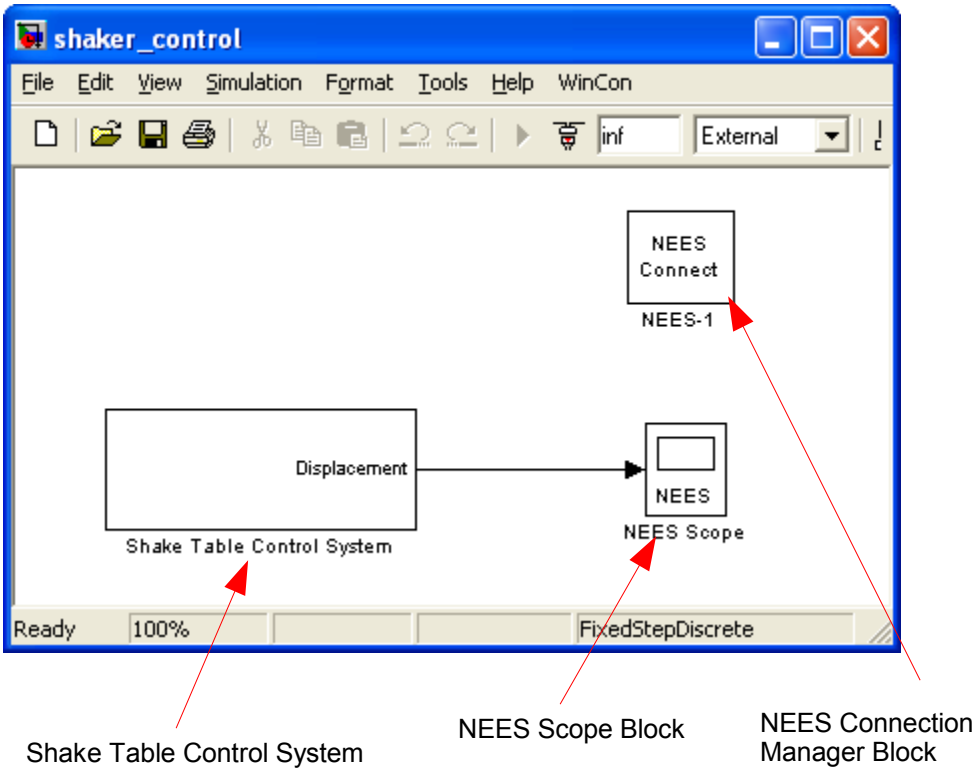


Figure 2: Components required to stream data from a Simulink diagram.

The diagram shown in Figure 2 shows the components required for streaming data from a Simulink diagram. The key components in the diagram are shown below.

- **Shake Table Control System**. This block contains a series of other blocks put together to control the shake table. These blocks are grouped together in what is called a subsystem. The subsystem is named the Shake Table Control System. There is a single output from this subsystem; **displacement**. The displacement parameter is the measured position of the shake table.
- **NEES Connect Block**. The NEES Connect Block manages the connection parameters for the other NEES blocks. It is with this block that the remote connection settings made. Referencing this block to manage the other NEES block's connection settings makes it simple to adjust the connection parameters globally.
- **NEES Scope Block**. The NEES Scope block is responsible for streaming data to the WinCon Web Daemon. The connection parameters for this block may be entered individually, or it may reference the NEES Connect block to manage the connection parameters.

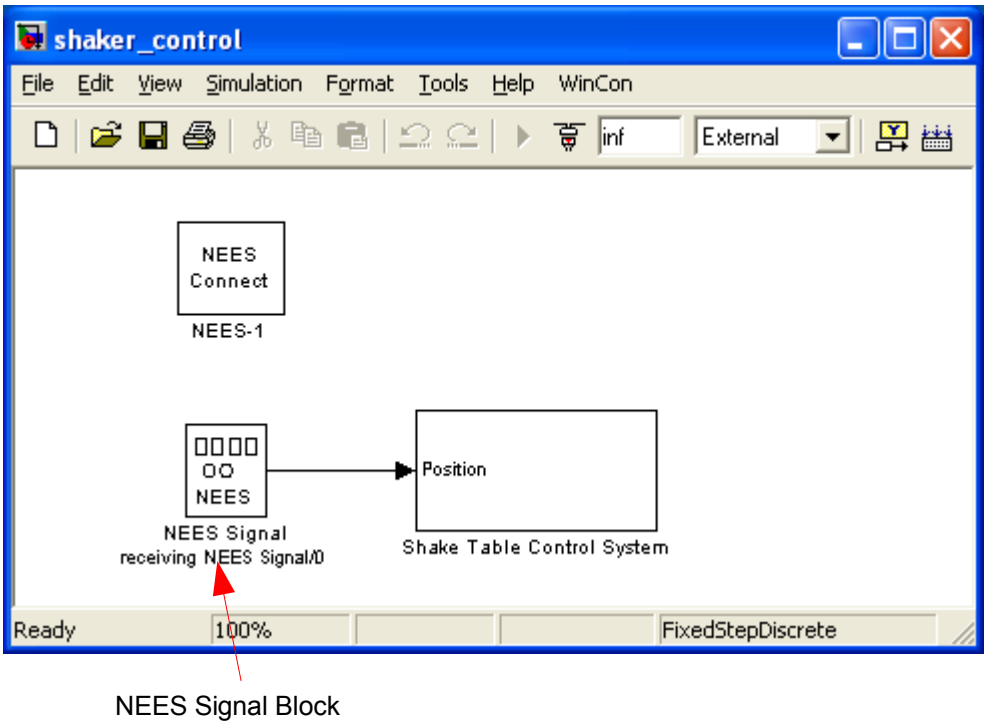


*Figure 3: Components required to send data **to** the Simulink diagram.*

The diagram in Figure 3 shows the components necessary to send data to the Simulink diagram. The components are identical to Figure 2 with the NEES Scope block being removed and the NEES Signal block being added. The NEES signal block may accept input values from the Data Turbine and input them as parameters to the Simulink diagram.

- **NEES Signal Block**. The NEES Signal Block accepts data from the Data Turbine and sends the data to the Simulink diagram.

NEES Connect Block

The parameters for the NEES Connect block are shown below.

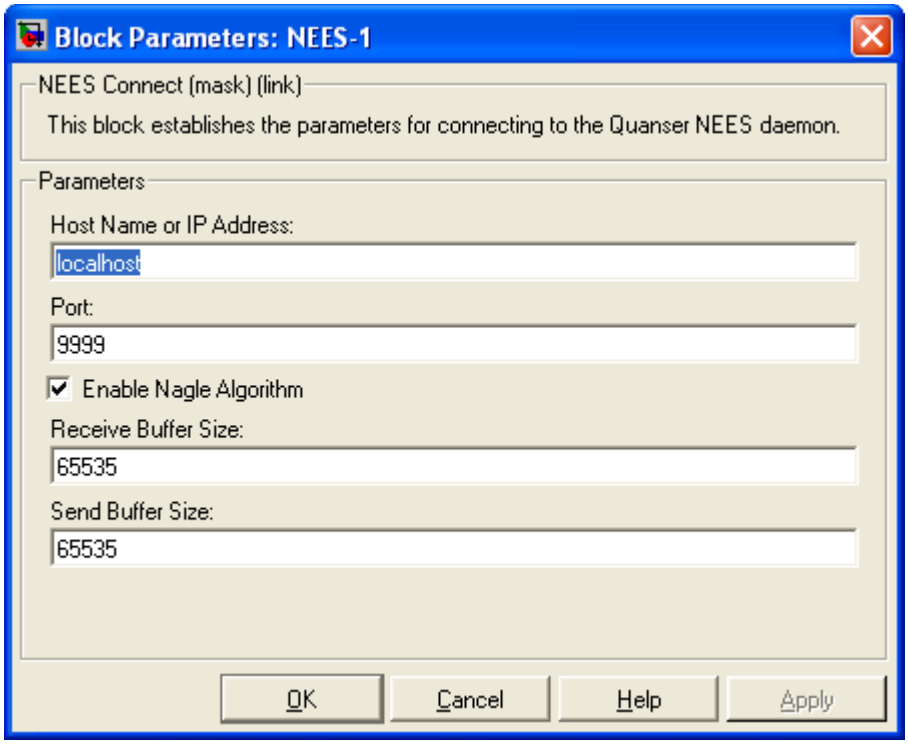


Figure 4: Block parameters for the NEES Connect block.

- **Host Name or IP Address**. This is the address of where the WinCon Web Daemon is located on the network. This information can be entered as a network address in the form of 192.168.1.1 or as a network name.
- **Port**. This represents the port on which The Web Daemon will listen for incoming connections from the NEES Scope and NEES Signal blocks.
- **Enable Nagle Algorithm** enables the use of the Nagle algorithm which buffers small packets to improve network efficiently by reducing the number of packets sent.
- **Receive Buffer Size**. The size of the receive (receive) buffer in bytes to use. Data is buffered to allow for network congestion. The default values should suffice for most applications.

- **Send Buffer Size**. The size of the transmit (send) buffer in bytes to use. Data is buffered to allow for network congestion.
-

NEES Signal Block

The parameters of the NEES Signal block are described below.

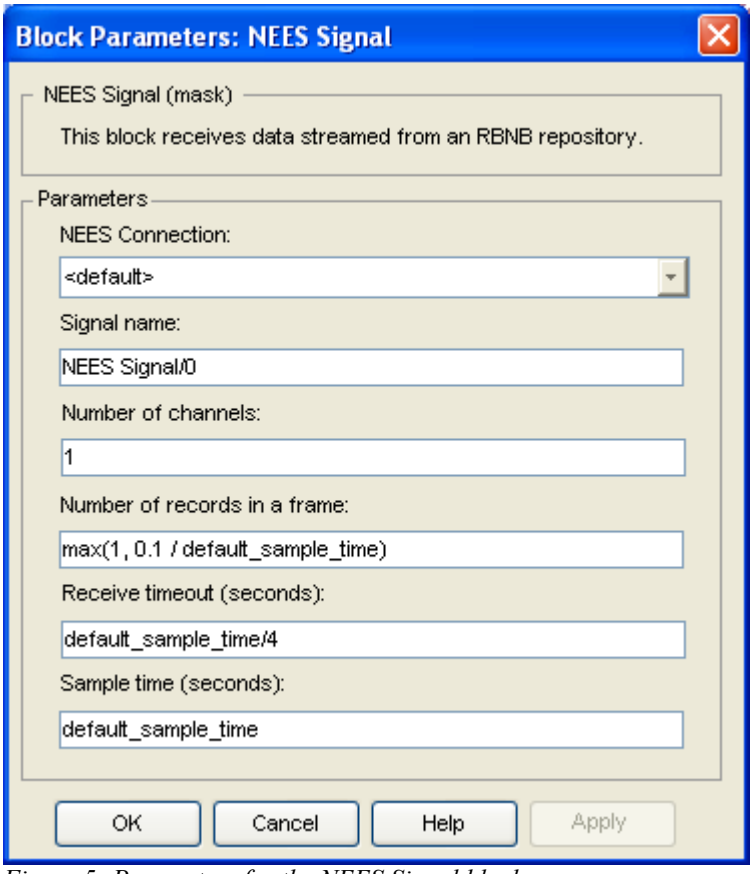


Figure 5: Parameters for the NEES Signal block.

- **NEES Connection**. This specifies how the connection parameters are managed. By selecting <default> the NEES Connect block is selected. Multiple NEES Connection blocks may be used in the diagram to stream data to different locations, so this dialog allows the user to select which block is in use.
- **Signal Name**. The name of the signal which is in the Data Turbine. Note that this must be the fully qualified name of the signal. The signal is retrieved by name reference from the Data Turbine.
- **Number of Channels**. Specifies the number of channels used by this block.
- **Number of Records in a Frame**. Specifies the number of records (samples) in a frame. The value is entered as a formula by default. The value of default_sample_time is typically 0.001s, so there will be 100 records stored within a single frame.

- **Receive timeout**. This value specifies how long to wait before issuing a timeout. A higher value may be required when there is a significant amount of network traffic.
- **Sample Time**. This value represents the time between samples. By default, this is set to the default sample time (0.001s).

NEES Scope Block

The parameters of the NEES Scope block are described below.

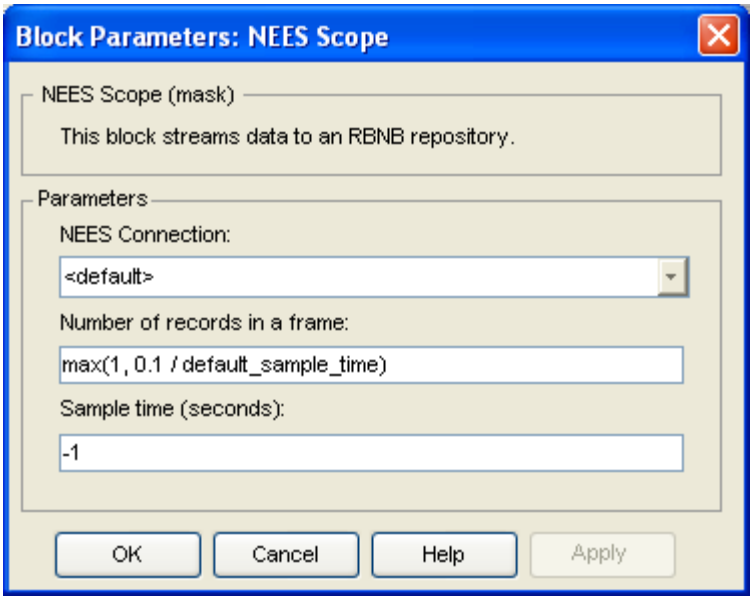


Figure 6: Parameters for the NEES Scope block.

- **NEES Connection**. This specifies how the connection parameters are managed. By selecting <default> the NEES Connect block is selected. Multiple NEES Connection blocks may be used in the diagram to stream data to different locations, so this dialog allows the user to select which block is in use.
- **Number of Records in a Frame**. Specifies the number of records (samples) in a frame. The value is entered as a formula by default. The value of default_sample_time is typically 0.001s, so there will be 100 records stored within a single frame.
- **Sample Time**. This value represents the time between samples. By default, this is set to the value of -1. In Simulink, this instructs the block to use the sample time of the input to this block. In other words, if the preceding block uses a sample time of 0.001s, this block will inherit this same sample time.

SOFTWARE PREREQUISITES

This software relies on several software components. At the core of this software is the real-time software WinCon, by Quanser Inc. **WinCon version 5.0.18.0** or higher is required to run the software (Java Daemon). In addition, the TCP/IP communications blocks are required to properly run the software.

A typical setup includes the following components:

- WinCon **5.0.18.0** (or higher) with TCP/IP blocks installed.
- MATLAB 7.0.1
- RTX 6.0.1
- Microsoft Visual Studio 6.0 or .NET editions with latest service packs

For detailed information on the setup and usage of WinCon, please refer to the WinCon user manual. This manual covers the required software components and the installation process in detail.

In addition to the software mentioned above, the WinCon Web Daemon software requires a Java Virtual Machine (JVM) to run. The software has been tested with Java 5 from Sun. This software s available for download from the Sun website: http://java.sun.com/

The user may choose to use either the Java Runtime Environment (JRE) or the Java Development Kit (JDK) versions of the software. If you intend to do any Java programming, the JDK download should be selected. To just run the software, on the JRE is required.

Software Requirements Checklist

The required software components are listed here for convenience. The user should still consult the proper manual for additional setup and usage instructions. As well, be aware that the ***WinCon compatibility chart should be consulted before deviating or upgrading any of the components listed below.***

- ✔ Java 5. Either the **JRE** (runtime environment) or **JDK** (development platform). Download the appropriate package from http://java.sun.com/
- ✔ WinCon 5. Build version 5.0.18.0 or later, **with TCP/IP support**.
- ✔ MATLAB 7.0.1
- ✔ RTX 6.01
- ✔ Microsoft Visual Studio 6.0 or .NET version, with latest service packs.
- ✔ Windows XP or 2000
- ✔ Creare RBNB Data Turbine. Download from http://outlet.creare.com/rbnb/

RUNNING THE SOFTWARE

Note: In order to run the software, the user will require a running RBNB Data Turbine to connect to. This may be running on the user's local machine, or it may be a remote address. The address of this machine running the Data Turbine and the connection port must be known.

The WinCon Web Daemon software is packaged as an executable jar file. This means that on most Windows based systems, the user may double-click to execute the program. This should work correctly for Java 5 installations. *A Java 5 Virtual Machine (JVM) must be installed on the system*. When the user double clicks the file, the following window should be visible.

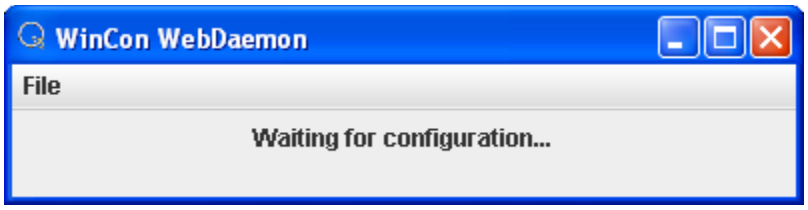


Figure 7: Opening window of the WinCon Web Daemon.

Note the message indicating that the program is waiting for configuration information. Without the configuration information, the location of the Data Turbine is unknown. There are several methods to configure the WinCon Web Daemon These methods are listed below, and then discussed in further detail.

- Configuration from within the application
- Using a defaults.xml file
- Opening a previously saved configuration file

The WinCon Web Daemon uses an XML file format for storing the configuration data. This makes editing and using the configuration files simple and easy to maintain.

Configuration From the Application

Using the menu in the application, select File->Configure... This will open the configuration dialog. The configuration dialog will appear and allow the user to make changes to the way the WinCon Web Daemon works.

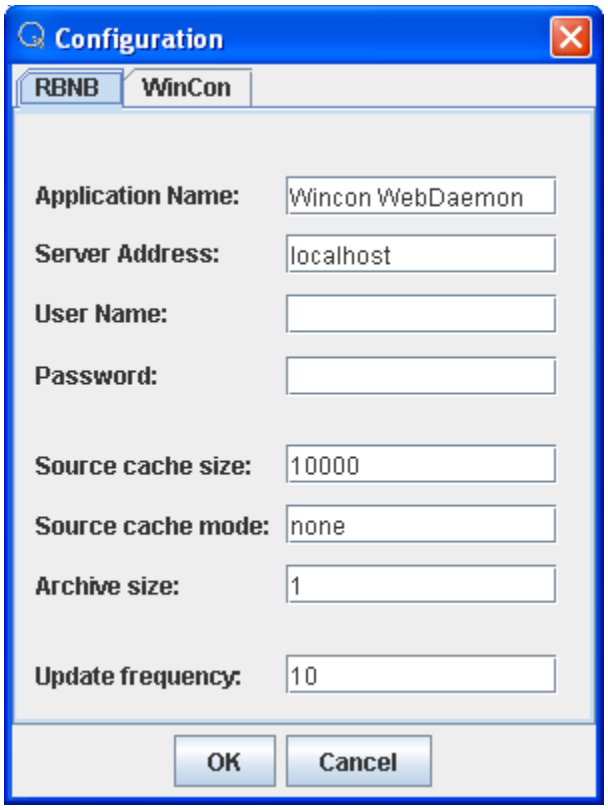


Figure 8: The RBNB connection configuration dialog.

The configuration dialog contains two information panes, one for RBNB configuration and one for WinCon configuration.

- **Application Name**. This is the name of the source as it will appear in the Data Turbine's tree view. You may rename this to something more meaningful if desired. The name is in the form of a string, and may contain, characters, numbers, and spaces.
- **Server Address**. The server address is the network address of the RBNB Data Turbine. This can be entered in a variety of ways including the formats,

  192.168.1.1 (IP address)

  192.168.1.1:3333 (IP address and port number)

  data_turbine_server (network name)

  data_turbine_server:3333 (network name and port number)
- **User Name**. The assigned user name. Note that on some distributions of Data Turbine this functionality may be limited.
- **Password**. The assigned user password. Note that on some distributions of Data Turbine this functionality may be limited.
- **Source cache size**. The size of the cache used for storing data sent to the Data Turbine. This is effectively the amount of data that may be temporarily stored on the Data Turbine before it is overwritten. Note that on some distributions of Data Turbine this functionality may be limited.

- **Source cache mode.** The mode to be used for archiving data. Note that on some distributions of Data Turbine this functionality may be limited.
- **Archive size**. The size of the archive used for permanent storage. Note that on some distributions of Data Turbine this functionality may be limited.
- **Update frequency**. The frequency that data will be written to Data Turbine. Effectively, the Web Daemon may buffer data for a short period of time to optimize network performance. The units are in Hz, with the default being 10 Hz.

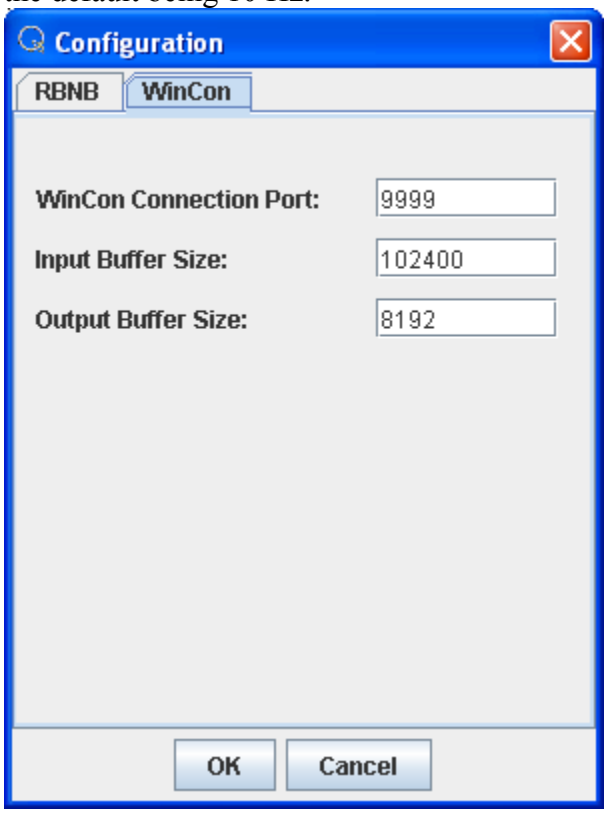


Figure 9: WinCon connection configuration dialog.

- **WinCon Connection Port**. The application will listen for incoming connections from WinCon on this port. Note that if this value is changed, the connection parameters of the individual NEES Scope blocks will need to be changed. For more information see the section titled Simulink Blocks.
- **Input Buffer Size.** The buffer size (in bytes) to use for buffering incoming data. This may be adjust based on sampling rate and network congestion.
- **Output Buffer Size**. The buffer size to use for buffering outgoing data.

To accept any changes and to re-connect to the Data Turbine, click OK when finished. To cancel any changes, click on the Cancel button.

Note that when OK is selected, the connection to Data Turbine will be closed and re-opened automatically for the changes to take effect.

When the dialog is closed by clicking OK, WinCon Web Daemon will close the open network connection with the Data Turbine. ***For this reason it is recommended to stop the controller running on WinCon first, before making any changes to the WinCon Web Daemon settings.***

Once the connection has been made, the message in the application window will read, Daemon running."

Figure 10: The "Daemon running" message displayed when connected to Data Turbine.

Using the Defaults.xml File

Using the defaults.xml file allows the user to setup a configuration whenever the application is launched. Essentially the defaults.xml file allows the user to change configuration information in much the same way as the Configuration Dialog, but with more options and flexibility.

Without going into details about the format of the file, the relevant fields to edit are noted below. For convenience, a listing of the defaults.xml file is reproduced here.

```
<?xml version="1.0" encoding="UTF-8"?>
<java version="1.5.0_04" class="java.beans.XMLDecoder">
 <object class="com.quanser.neesit.daemon.DaemonCore">
  <void property="applicationName">
   <string>Wincon WebDaemon</string>
  </void>
  <void property="connectionThreadSleep">
   <int>100</int>
  </void>
  <void property="dataDuration">
   <double>5.0</double>
  </void>
  <void property="fetchTimeout">
   <long>1000</long>
  </void>
  <void property="inputStreamBufferSize">
   <int>102400</int>
  </void>
  <void property="outputStreamBufferSize">
   <int>8192</int>
  </void>
  <void property="password">
   <string></string>
  </void>
```

```
  <void property="serverAddress">
   <string>localhost</string>
  </void>
  <void property="sourceArchiveSize">
   <int>1</int>
  </void>
  <void property="sourceCacheMode">
   <string>none</string>
  </void>
  <void property="sourceCacheSize">
   <int>1</int>
  </void>
  <void property="threadIdleTime">
   <int>100</int>
  </void>
  <void property="useUTC">
   <boolean>true</boolean>
  </void>
  <void property="userName">
   <string></string>
  </void>
 </object>
</java>
```

The format of the file follows a property / value pairing. For each property, there exists a value. To edit the file, simply locate the property (shown in bold) and then edit the corresponding value. Save the file, and then open it (or specify it as the name defaults.xml).

As with the configuration dialog, the names of the properties are quite descriptive. They are listed below, but the user should note that the file is case sensitive. Only the most commonly used properties are described here. Consult the appendix for a full explanation of the remaining properties.

- **applicationName**. This is the name of the source as it will appear in the Data Turbine's tree view. You may rename this to something more meaningful if desired. The name is in the form of a string, and may contain, characters, numbers, and spaces.
- **serverAddress**. The server address is the network address of the RBNB Data Turbine. This can be entered in a variety of ways including the formats,

  192.168.1.1 (IP address)

  192.168.1.1:3333 (IP address and port number)

  data_turbine_server (network name)

  data_turbine_server:3333 (network name and port number)
- **userName**. The assigned user name. Note that on some distributions of Data Turbine this functionality may be limited.

- **password**. The assigned user password. Note that on some distributions of Data Turbine this functionality may be limited.
- **sourceCacheSize**. The size of the cache used for storing data sent to the Data Turbine. This is effectively the amount of data that may be temporarily stored on the Data Turbine before it is overwritten. Note that on some distributions of Data Turbine this functionality may be limited.
- **sourceCacheMode.** The mode to be used for archiving data. Note that on some distributions of Data Turbine this functionality may be limited.
- **sourceArchiveSize**. The size of the archive used for permanent storage. Note that on some distributions of Data Turbine this functionality may be limited.
- **threadIdleTime**. The frequency that data will be written to Data Turbine. This is actually expressed as f = 1000/(threadIdleTime), where the time is expressed in units of milliseconds [ms]. For example, a threadIdleTime of 250 would provide an update rate of 1000/250 = 4 Hz.
- **listenPort**. The application will listen for incoming connections from WinCon on this port. Note that if this value is changed, the connection parameters of the individual NEES Scope blocks will need to be changed. For more information see the section titled Simulink Blocks.
- **inputStreamBufferSize.** The buffer size (in bytes) to use for buffering incoming data. This may be adjust based on sampling rate and network congestion.
- **outputStreamBufferSize**. The buffer size to use for buffering outgoing data.

Opening a Configuration File

Selecting File→Open will open a file chooser dialog to allow the user to select configuration files to open. This has a similar effect as putting the configuration settings in defaults.xml with the exception that this file may be loaded at any time, not just when the application starts.

Another method of opening a configuration file is by specifying the filename on the command line. An example of this follows.

```
java -jar wwdaemon.jar configuration.xml
```

In this example, the configuration file to be loaded is called configuration.xml. To change the loaded file, change this filename. The rest of the command line should remain the same. Specifying the configuration in this manner launches the application with the settings specified in the configuration file.

Saving a Configuration File

Selecting File→Save will open a dialog to allow the saving of the configuration files. These files may then be loaded with the File→Open command, from the command line, or by default by naming the file defaults.xml and saving it in the applications working directory. (the directory where it is run from)

Creating a New Defaults.xml file

In the event that the defaults.xml file is lost or the user wish to use the default settings, a defaults.xml file may be created. The required settings may be typed into a file and then saved. A simpler method would be to have the application generate the defaults for you. To do this, follow these steps.

1. Delete or rename the existing defaults.xml file. When the application is launched, it will not find this file, and setup with the initial default values.
2. Start the application, without any file arguments. This will ensure that no file is loaded.
3. Select File→Save from the menu. Type in the name **defaults.xml**. This will create a new defaults.xml file for you.